Performance Analysis of a Neuro Fuzzy Algorithm in Human Centered & Non-Invasive BCI

Timothy Scott C. Chu[1,2], Alvin Chua[2] and Emanuele Lindo Secco[1] [0000-0002-3269-6749]

[1] Robotics Laboratory, School of Mathematics, Computer Science & Engineering, Liverpool Hope University, UK
[2] Mechanical Engineering Department, De La Salle University, PH
19009445@hope.ac.uk, alvin.chua@dlsu.edu.ph,seccoe@hope.ac.uk

Abstract. Brain-Computer Interface can be non-invasive devices that obtain signals generated from the brain and are then manipulated to suit various applications. A popular application for BCI is interfacing with robotics; and, each BCI – Robotics system employed different Machine Learning algorithms. This study aimed to present a performance analysis for a Neuro-Fuzzy algorithm, specifically the Adaptive-Network-Fuzzy-Inference System (ANFIS), to classify EEG signals retrieved by the Emotiv INSIGHT. An SVM algorithm is also developed to serve as a reference vs the ANFIS's performance. A methodology for generation and acquisition of EEG signals can be used by researchers as reference. Facial and Eye Gestures were utilized as means of EEG signal generation which are fed to both algorithms for simulation experiments. Results showed that the ANFIS tend to be more reliable and marginally better than of the SVM algorithm. Compared to SVM, the ANFIS took significant amounts of computational resources requiring higher specs and training time.

Keywords: BCI, BMI, human-centered interface, ANFIS, SVM.

1 Introduction

One idea from science-fiction movies that pique the interest of its audience is the ability to control things with the use of their mind. However, that idea may be a reality today with the developments in *Brain-Computer Interfaces (BCI)* and *Artificial Intelligence*. The human brain interacts with limbs by coursing through electric signals along the nerves and synapses that serve as the bridge connecting the brain to every part of the body. These electrical activities can be captured by Brain-Computer Interfaces (BCI) machines by conducting a test called electroencephalogram (EEG) [1]. This test enabled medical professionals to observe and detect anomalies in a patient's brain. BCI machines can either be invasive or non-invasive. When surgical procedures are required to strategically embed a sensor inside the head of an individual, this is considered to be an invasive BCI machine. If the sensor is positioned outside of the body, whether by touching the scalp of an individual or have some distance to the individual, this is considered to be a non-invasive BCI machine. Developments in BCI technology provided

researchers not only access to affordable BCI machines, but also opportunities in robotic developments through the integration of robotics into BCI systems.

Researchers from the University of Dayton, Ohio conducted a study that used Brain-Machine Interface to control a robotic arm called *Robai Cyton Veta Robotic Arm*. The researchers used the *Emotiv Epoc (EPOC)* headset to extract EEG signals from the brain using the developer's testbench [1]. The study utilized a *Linear Discriminant Analysis (LDA)* which functioned as a classifier for the obtained EEG signals. The process began when a user was asked to execute gestures on each hand. With the executed gestures, EEG signals were generated, captured by the EPOC, and were pre-processed by visually analyzing the signals to determine the location of spikes from the recorded data of a particular gesture instance. The information obtained from multiple instances were collated to form a dataset of features and targets that reflected EEG features of each gesture. This method allowed direct utilization of raw EEG data for implementation instead of the usual method of filtering obtained signals. The proposed new method of data classification in this study was able to successfully distinguish separate sets of actions such as right, right neutral, left, and left neutral with an accuracy of close to 100%. Additionally, the researchers were also successful in controlling the robotic arm. The dataset utilized by the proponents was downscaled to minimize complexity and the number of gestures utilized was also reduced to 3. The proponents of the research noted that the obtained accuracy data may vary when introduced with larger datasets and more degrees of freedom or features.

Researchers from AGH University of Science and Technology, Poland were able to control a *Lego Mindstorms robot* and conduct pick and place exercises with the use of the *EPOC* headset [2]. The data was gathered through the *Emotiv SDK* and the raw EEG signals were then processed in another software called *LabVIEW*. The researchers used the *Steady State Visual Evoked Potentials (SSVEP)* technique where visual prompts were tied to a certain action command. This was achieved by creating a LED panel with 4 colors, blue, red, white, and green and each color was set to flicker to the following frequencies: f_{blue}= 28 Hz, f_{red}= 30 Hz, f_{white}= 32 Hz, and f_{green}= 34 Hz. Under the LabVIEW environment, the researchers used *Power Spectral Density (PSD)* for feature extraction and a machine learning algorithm called the *Classification Tree* method (specifically *ITR*) was used. The researchers opted for the ITR learning algorithm as it was a popular in BCI system implementations. The objective of the research was to prove that it was possible to construct a relatively robust and low-cost BCI system to manipulate mechanical devices. After the testing, the system developed by the researchers was successful with an average effective rate of 73.75%. Researchers showed that having visual prompts allow for easier use and tracking of executable commands when using BCI technology.

Researchers from Sichuan University, China made a study on the performance of *Convolutional Neural Network* (*CNN)* algorithm when fed with data filtered from the *Random Forest (RF)* algorithm [3]. The experiment made use of an online EEG dataset that represented different hand gestures and its corresponding signals in time series. The data was imported into the system and was screened using the Random Forest

algorithm to remove any unnecessary data. During the experiment, it was found that the RF algorithm was accurate and robust in managing various dataset types. Upon filtering the 72000 x 16 dataset, the researchers then proceeded to process the data by denoising it with *Savitzky-Golay filtering*. Finally, the processed data is fed to a modified CNN algorithm based on *AlexNet* with a total of 5 convolutional layers, 3 fully connected layers, and 2 pooling layers. The task of the algorithm was to classify the processed dataset into 6 categories of hand actions and the output was counter-checked with actual outputs based from the data. This experiment was set to run for 2 iterations with different ratios of training set over the total set. The average accuracy for the 2 iterations were 93.22% and 93.01% respectively. In their documentation, the researchers mentioned that there was some interference in the signals as every hand action is not entirely unique; that there was a degree of overlap in the signal readings with other actions. This affected the result of some specific hand action causing its accuracy percentage to be relatively low. However, the proposed technique of filtering with the RF algorithm, denoising of the data, and utilizing the CNN algorithm was still able to return accuracy readings above 90%.

In addition, various study that integrated BCI to robotics were conducted like the study conducted in [4] which served as a viability study which validates the applicability of Emotiv Neuroheadset in robotic systems. Together with studies [5, 6], they all utilized the algorithms in the Emotiv provided software development kit (SDK). Apart from the 2 sets of authors who utilized mental commands to control their respective robotic systems, the researcher in [5] explored a different approach by using the Left and Right wink face gesture to control a robotic arm; additionally, they were able to highlight 2 EEG sensors, AF3 and AF4, which were significant in obtaining EEG information for the said gesture. Finally, the study conducted in [7] also utilized the concept of SSVEP as their stimulus together with the *Fast Fourier Transform Spectral Analysis* as their classifier. The study was able to perform well with an accuracy rating of 92.5%, but its limitation is found in the commands it executed. The study operated a quadcopter with the use of the Emotiv Neuroheadset and moved along the experiment space with a combination of pitch and yaw commands. Additionally, since their study is limited with 4 commands, the application did not utilize the full potential of the 6 degrees-of-freedom (D.O.F) of the drone.

Individuals who possess physical limitations experience a degree of restriction from the usual lifestyle. This limitation may hinder them from performing an important task either at home or in their work. This can be addressed by developing a BCI-Robotics system as an attempt for individuals to regain their lifestyle. However, in the field of BCI control implementation, the distinguishing factor between studies is the performance of the algorithm used in the application. Widely used algorithms are usually *Convolutional Neural Network* or *Artificial Neural Network*, these algorithms offer high accuracy however take up a significant amount of computational time [8]. Another common algorithm that is being utilized in this field is the use of *Support Vector Machines (SVM)*; this algorithm can effectively classify signals efficiently, given that a proper kernel is provided by the user to create effective hyperplanes, often this becomes

a huge challenge to inexperienced users [9]. Also, the SVM's performance drops according to the size of the dataset, consequently taking up more computational time. The challenge is to utilize or develop an algorithm that will offer a good balance between accuracy and computational time; and, should be capable to accompany sufficient input information. The integration of BCI and Machine Learning to robotics can offer massive benefits to society. A particular study noted that BCI implementation can enable people who possess physical limitations, execute everyday tasks as if the limitation wasn't present. This idea was suggested in an article wherein a BCI machine is used to control computer peripherals such as the mouse and keyboard [10]. Consequently, BCI also has the potential of replacing traditional hand-held controllers as it is also capable of controlling RC vehicles. In one of the presented studies, it mentioned that a degree of overlap with the EEG signal features between gestures affected the performance of the machine learning algorithm. With this consideration, an algorithm capable of handling this overlap would be well suited in this application; and one of these algorithms is the *Adaptive-Neuro-Fuzzy Inference System (ANFIS)*. The ANFIS algorithm is one form of *Fuzzy Neural Network*, a hybrid between Fuzzy Logic and Neural Networks, that utilizes a first-order Sugeno model [8]. This research explores the viability of the ANFIS algorithm in BCI interface in conjunction with the Support Vector Machines (SVM), where the latter algorithm serves as a baseline comparison to put in perspective how well the ANFIS performs on a number of parameters such as accuracy. In addition, this research aims to contribute literature on the performance analysis of Neuro-Fuzzy and SVM algorithms and its potential implementation to BCI control systems. This would provide readers some insights on the collection and processing of the raw EEG data; then using the obtained information to drone applications or their desired robotics application. The output of this research may be fine-tuned and be utilized by individuals with physical disabilities for a desired purpose.

2 Theories Involved

2.1 Brain and Brain Rhythms

Brain. The brain is composed of multiple parts; however, this research focuses on the general function of the different parts of the Cerebral Cortex. The Cerebral Cortex has 4 lobes with specialized functions as shown in Figure 1. The Frontal Lobe is considered to be the motor portion of the brain since it manages motor skills and other cognitive functions. Action tasks that require muscle movements are controlled by this lobe. This lobe also performs associative processes such as learning, thinking, and memory. As the name suggests, the Frontal Lobe is located at the front portion of the brain. The Parietal Lobe is the portion of the Cerebral Cortex that manages all somatosensory feedback from the body. This lobe is positioned on the top of the brain, behind the Frontal Lobe. The Temporal Lobe, located below the Frontal and Parietal Lobes of the brain, is in charge of processing the auditory stimulus. Finally, the Occipital Lobe is positioned at the back of the brain. It processes and manages visual

information obtained directly from the eyes [11]. Table 1 shows a summary of the different parts of the Cerebral Cortex and its respective functions.

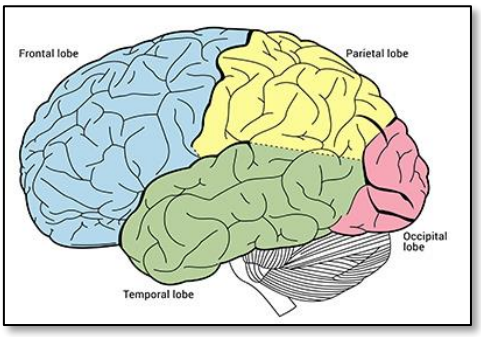

Fig. 1. Lobes of the Brain, obtained from [12]

Table 1. Parts of the Brain and Functions

Brain Lobe	Function
Frontal Lobe	Motor Functions, Cognitive Functions
Temporal Lobe	Processes Auditory Input, Emotions
Parietal Lobe	Processes Somatosensory Feedback
Occipital Lobe	Processes Visual Input

Brain Rhythms. One way to describe the behavior of the brain is by observing and measuring its brainwaves. Brainwaves are electrical signals which form non-linear patterns that operate on various frequencies depending on the certain conditions, e.g. relaxed state, excited state, resting state, etc. [13]. There are 5 categories of Brainwaves or Brain Rhythms, the Delta Waves, Theta Waves, Alpha Waves, Beta Waves, and Gamma Waves. Table 2 shows the different waves and its corresponding frequency range.

Table 2. Brainwaves and Corresponding Frequency Range

Brainwaves	Frequency Range
Delta Waves	< 4 Hz
Theta Waves	4 Hz – 7 Hz
Alpha Waves	7 Hz – 13 Hz
Beta Waves	14 Hz – 30 Hz
Gamma Waves	> 30 Hz

2.2 Band-Pass Filters

The research employs Band-pass filtering to remove noise in the data. Band-Pass filtering is a standard EEG pre-processing technique that uses the concepts of pass-band and stop-band. A pass-band allows information that is within the cut-off frequency to

go through the filter, while a stop-band rejects information that is beyond the cut-off frequency. Extending this concept gives 4 common filter types, Low Pass filters, High Pass Filters, Band-pass Filters, and Band-stop or Notch Filters. Figure 2 represents each filter in graphical form with frequency on the x-axis and amplitude on the y-axis.

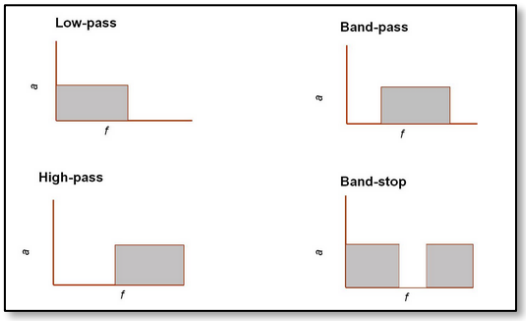

Fig. 2. 4 Common Types of Filters [14]

The shaded box in Figure 2 represents the pass-band while the unshaded areas represent the stop-band of the filter, this is also represented by Figure 3.

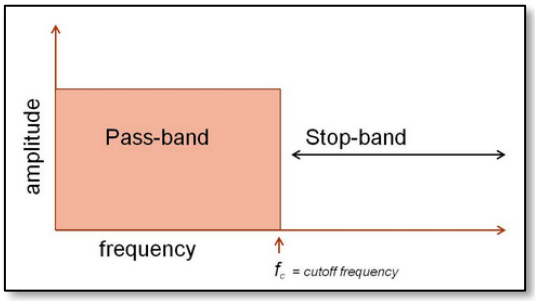

Fig. 3. Frequency Response of Filter Type [14]

The Low Pass filters allow information below the set cut-off frequency to pass through, which is good in removing high-frequency noises, such as information on the Gamma wave level. The High Pass filter on the other hand is the opposite of the Low Pass filter where information above the set cut-off frequency is allowed to pass through. Band-pass filters have 2 cut-off frequencies, the lower cut-off, and the higher cut-off frequencies; and, information inside the boundary defined by the 2 cut-off frequencies passes through the filter, rejecting information beyond the boundary. The Band-stop or Notch filter is the inverse of the Band-pass where information beyond the boundary is allowed to pass through the filter, rejecting information within the boundary set by the 2 cut-off frequencies.

If the obtained EEG signal can be filtered by square waves, then filtering signals would be easy and accurate. Unfortunately, this is not the case as EEG signals, when

plotted, tend to form sinusoidal waves. Consequently, hard cut-off frequencies could cost the user some important information. Butterworth Band-pass Filters extend the previously explained concept and determines the normalized gain, and instead of hard cut-off passes, it introduces the idea of gain into the system. The Butterworth filters introduce 2 transfer functions which are represented by Equations 1 and 2. The two equations refer to the High-pass filter and the Low-pass filter respectively.

$$|H(\omega)| = \frac{Ao}{\sqrt{1+\left(\frac{\omega_o}{\omega}\right)^{2n}}} \tag{1}$$

$$|H(\omega)| = \frac{Ao}{\sqrt{1+\left(\frac{\omega}{\omega_o}\right)^{2n}}} \tag{2}$$

Where $H(\omega)$ is the normalized gain, A_o as the max gain in pass-bands, ω_o is the cut-off frequency, lower for Low-pass filters (eq. 2) and higher for High-pass filters (eq. 1), ω is the frequency of the input signal, and n as the order of the filter. The Butterworth filters utilize a normalized approach in filtering the signal, generating a bell-like shape filter as shown in Figure 4.

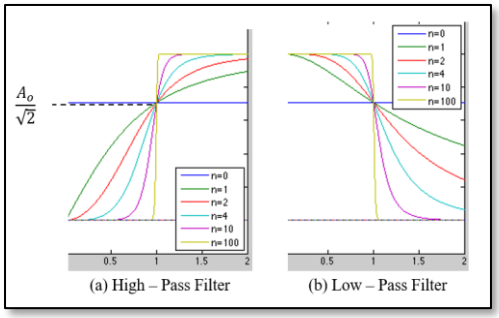

Fig. 4. (a) High-Pass (b) Low-Pass Butterworth Filters of different orders

For example, a cut-off frequency of 1 Hz on the 1st order is specified in a high-pass Butterworth Band-pass filter, shown in Figure 4 (a). One can observe that a parabolic shape is formed, represented by the green line; and, any information inside the established boundary is allowed to pass through. Extending the example, EEG signals on 1 Hz are introduced into the filter. The Butterworth filter transforms the input to its normalized form with Equation 1, consequently it holds:

$$|H(\omega)| = \frac{Ao}{\sqrt{2}} \tag{3}$$

2.3 Adaptive Neuro – Fuzzy Inference System (ANFIS)

JS Jan [8] developed a Neuro-Fuzzy algorithm called the Adaptive-Network-Fuzzy-Inference-System, or ANFIS for short. The model combines both the concept of Fuzzy Inference System and Adaptive Network, making this a hybrid model.

The Fuzzy Inference System is a core concept from Fuzzy Logic wherein it generally possesses 4 main functions namely, the (i) Knowledge Base, (ii) Fuzzification, (iii) Defuzzification, and the (iv) Decision – Making Unit as shown in Figure 5.

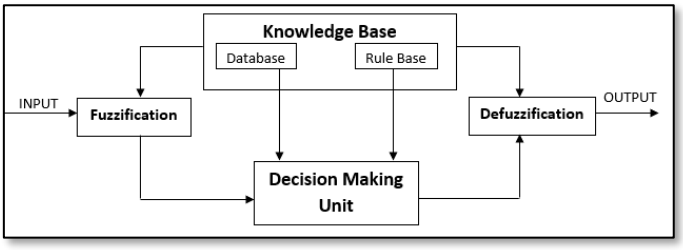

Fig. 5. Fuzzy Inference System [8]

The algorithmic sequence starts on the Knowledge Base, which holds the Database and Rule Base. This is where base information is extracted for Fuzzification, Defuzzification, and Decision-Making Unit. The Database is responsible for generating and storing membership functions which are used to interpret the data fed into the algorithm. The Rule Base is responsible for defining the occurrence of an outcome with respect to a set of criteria that takes the fuzzified values as input. Crisp values, which serve as inputs, are initially fuzzified, this is called Fuzzification, where crisp values are translated to membership values. The fuzzified input is then fed to the Decision-Making Unit where it runs the input to fuzzy operators (i.e. Min, and Max) and then compares them to a set of criteria established in the Rule Base to determine which is the appropriate output for the given input. However, the outputs of the Decision-Making Unit are still fuzzified values, to return them to crisp values, information from the Database is used to re-translate the values, this is called Defuzzification. An advantage of this system architecture is that it is capable of accepting vague inputs, also referred to as uncertain inputs, and return outcomes that are acceptable.

An Adaptive Network may be defined as a network of nodes, that comprises several layers. This network is responsible for the learning sequence of the algorithm; the objective of the Adaptive Network is to allow the algorithm to adapt to its mistakes and learn to increase the accuracy of the predictions made. This can be considered as a supervised learning algorithm, wherein design parameters are established and are integrated with the nodes to serve as efficacies; and, these efficacies are adjusted accordingly by means of training with a dataset to meet its objective. Various techniques are exercised in the training phase, one of the popular techniques is Gradient Descent. However, when this technique is implemented in large datasets, may possess high time complexity, consequently resulting in high computational costs. The ANFIS architecture combines the Fuzzy Inference System and the Adaptive Network. In addition to this architecture, the Adaptive Network is a combination of 2 different learning rules, which are usually, but not limited to, *Least Squares Estimates* (LSE) and the Gradient Descent.

The appropriate outcome is computed with a function (F) together with a set of input values ($\vec{I}$) and a set of design parameters (S), as represented in Equation 4. Incorporating the hybrid learning rule, the design parameters, S, can be collated and expressed in Equation 5.

$$output = F(\vec{I}, S) \tag{4}$$

$$S = S_1 + S_2 \tag{5}$$

Where S_1 represents the weights and thresholds of the hidden layer and S_2 represents the weights and thresholds of the output layer. Theoretically, the algorithm is subjected to two passes of learning rules to obtain necessary parameter values. Table 3 summarizes the learning process of the algorithm.

Table 3. Hybrid Learning Process for ANFIS Algorithm

	Forward Pass	Backward Pass
Premise Parameters	Fixed	Gradient Descent
Consequent Parameters	Least Squares Estimates	Fixed
Signals	Node Outputs	Error Rates

The explanation of the algorithm's architecture assumes a case scenario of two inputs and one output. The ANFIS algorithm normally possesses 5 layers; and, the initial layer is responsible for the fuzzification of the crisp inputs with some membership functions. Normally, a Gaussian membership function is utilized, however, other options such as triangular and trapezoidal membership function. The equation for the Gaussian membership function is expressed in Equation 6.

$$K(x_i, x_j) = e^{\frac{||x_i - x_j||}{2\sigma}} \tag{6}$$

Layers 2 and 3, are responsible for determining the parameters and normalized parameters using Equations 7 and 8 respectively; additionally, it is common to combine these two layers to a single layer, since both have similar functions and to save computational time.

$$w_i = \mu_{Ai}(x) \times \mu_{Bi}(x), i = 1,2 \tag{7}$$

$$\overline{w_i} = \frac{w_i}{w_1 + w_2} \tag{8}$$

On the 4th layer, consequent parameters are calculated with Equation 9. Generally, layers 2 up to 4 do the majority of the thinking by calculating a degree of the relation of each input value to a particular category.

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \tag{9}$$

Where O_i^4 is interpreted as the output of the i[th] node in layer 4. Finally, on the final thinking layer or 5th layer, all information from the previous layer is collated, calculated, and 'defuzzified' to produce a single output using Equation 10.

$$O_1^5 = \ overall\ output = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (10)$$

Figure 6 shows the sample ANFIS algorithm diagram with 2 inputs and a single output.

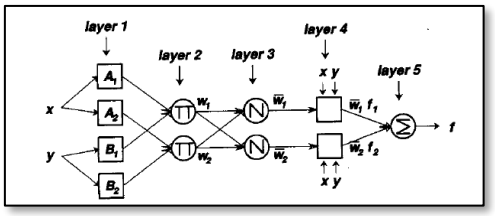

Fig. 6. ANFIS Diagram, 2 inputs, 1 output [8]

A simple ANFIS algorithm generally follows 2 rules that are shown in Equations 11 and 12 below and can be expanded appropriately to fit a particular use case.

$$Rule\ 1: If\ (x_1\ is\ in\ A_1) and\ (x_2\ is\ in\ B_1), then\ (\hat{y} = p_1 x_1 + q_1 x_2 + r_1) \quad (11)$$

$$Rule\ 2: If\ (x_1\ is\ in\ A_2) and\ (x_2\ is\ in\ B_2), then\ (\hat{y} = p_2 x_1 + q_2 x_2 + r_2) \quad (12)$$

Where x_1 and x_2 are the values inputted in the algorithm, and A_i and B_i are the fuzzy sets of the data. The crisp output $\hat{y}$ results in a value corresponding to the input values and is computed together with the design parameters p_i, q_i, and r_i.

3 Materials and Methods

Figure 7 shows the methodology flowchart implemented in this research. The process begins with the *Generation of EEG Raw Signals*. The generated signals are then *Retrieved and Transferred* by the *Emotiv INSIGHT* neuroheadset to the computing hardware to the *OpenViBE* software. In OpenViBE, the data will be *processed and recorded* into CSV files which will be sent to the Machine Learning algorithms for training and evaluation.

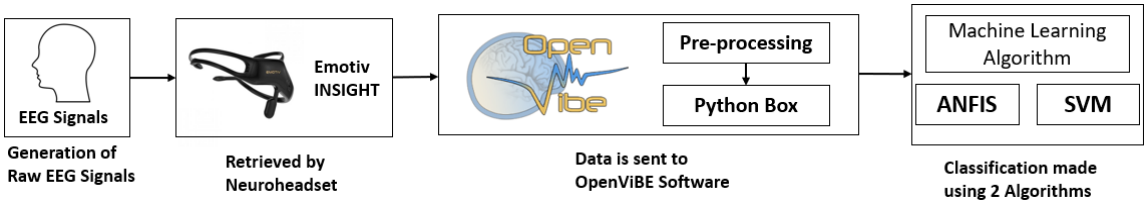

Fig. 7. Methodology Flowchart

3.1 Generation of EEG Raw Signals

The generation of EEG data was achieved with 2 methods; the first was by making and holding a face gesture for 15 seconds, this serves as the 1st EEG Dataset. These face gestures used were Neutral, Smile, Shocked, and Clench as shown in Figure 8.

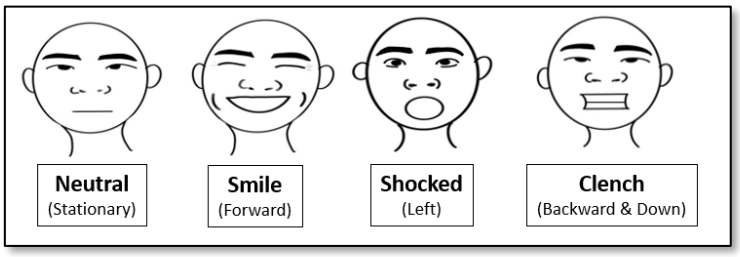

Fig. 8. *Face* Gestures

The second method of obtaining EEG data this time utilized eye gestures, such as Neutral, Eyes Widen, Left Wink, Right Wink, and Closed. In addition, instead of holding the gesture, instances are obtained, the mentioned gestures are shown in Figure 9.

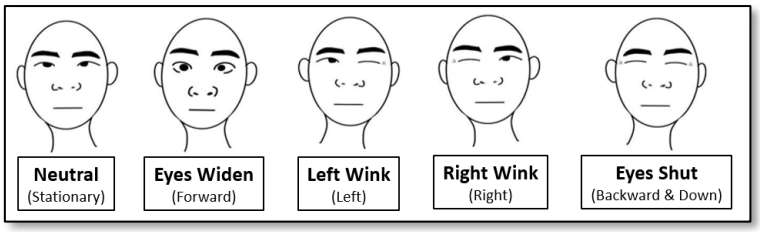

Fig. 9. *Eye* Gestures

This was considered to be the 2nd EEG Dataset. By doing the mentioned face gestures, brainwave activities were generated on the Frontal Lobe, which was detected and captured by a BCI Machine.

3.2 Retrieving and Transferring of Generated EEG Data

The Emotiv INSIGHT is a non-invasive *Brain-Computer Interface* (BCI) machine equipped with 5 electrode sensors that follow the international 10-20 system of electrode placements. Electrodes *AF3* and *AF4* are positioned by the Frontal Lobe of the brain, *T7* and *T8* are positioned by the Temporal Lobe of the brain, and *Pz* is positioned by the Parietal Lobe of the brain, shown in Figure 10. Electrode names are conveniently labeled to where it would be located, *F* stands for Frontal Lobe, *T* stands for Temporal Lobe, *P* stands for Parietal Lobe, *C* stands for Central Lobe, and *O* stands for Occipital Lobe.

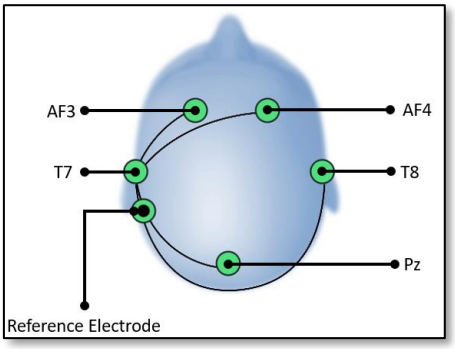

Fig. 10. Emotiv INSIGHT Layout

The neuroheadset obtains the generated EEG signals and translates it into values that can be easily understood by the researcher. With the method utilized in generating EEG data, sensors *AF3* and *AF4* were observed to be most effective. The values are measured in millivolts, and when plotted against time in a cartesian plane, it generates a visualization of a particular EEG signal similar to Figure 11.

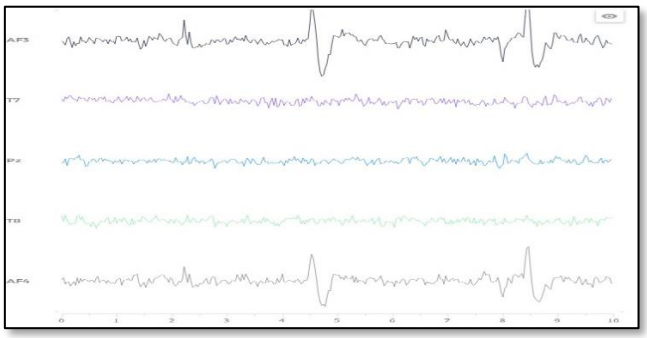

Fig. 11. EEG Signals

The INSIGHT has a sampling frequency of 128 Hz, which means that the neuroheadset obtains 128 samples of brain activity measured in millivolts for each electrode in a second. The measured values are sent to the computing hardware with the use of a dedicated Bluetooth dongle which is plugged in the computer upon use.

3.3 Pre-Processing of Data and Feeding to Algorithm

These data were then sent to the OpenViBE software where it was processed and recorded as a CSV file. OpenViBE is an acronym which stands for *Open Virtual Brain Environment*; and, it is open-source software that allows users to design and test BCIs. Additionally, the software is capable of processing EEG data in real-time as well as data streaming.

On the 1st EEG dataset, the research did not employ any filtering process to determine the capacity of the algorithms in managing raw EEG data. While on the 2nd EEG dataset, a *Temporal Filter* was applied, specifically a *5th Order Butterworth Band-Pass filter*. The filter blocked brainwaves under the Delta, Theta, and Alpha waves, focusing on the Beta and Gamma brainwaves. This was achieved by setting the lower cut-off and higher cut-off frequencies of the band-pass filter to be 13 Hz and 43 Hz, respectively, this serves as the 2nd EEG Dataset. The blocked brainwaves were observed to be sensitive to small movements, consequently generating noise in the dataset and affecting the overall performance of the algorithm. The filtered data is then recorded as a CSV file and was further processed heuristically by locating the minimum and maximum values on the sensors *AF3* and *AF4* of each instance. This approach served as an extension of the research conducted in [5], wherein the proponents of the mentioned researches used eye gestures to control a robotic hand. The generated datasets were then fed to the algorithms for training and evaluation.

Table 4. Datasets Generated and Used

Dataset	No. of Features	No. of Classifications	No. of Row Inputs	Total Array Size
2 Gesture Classification				
1st EEG Dataset (Face Gesture) 1 Sample Count	5	2	200	200 x 5
1st EEG Dataset (Face Gesture) 2 Sample Count	10	2	200	200 x 10
1st EEG Dataset (Face Gesture) 4 Sample Count	20	2	200	200 x 20
2nd EEG Dataset (Eye Gestures) 1 Sample Count	5	2	100	100 x 5
2nd EEG Dataset (Eye Gestures) 2 Sample Count	10	2	100	100 x 10
All Gesture Classification				
1st EEG Dataset (Face Gesture) 1 Sample Count	5	4	800	800 x 5
1st EEG Dataset (Face Gesture) 2 Sample Count	10	4	800	800 x 10
1st EEG Dataset (Face Gesture) 4 Sample Count	20	4	800	800 x 20
2nd EEG Dataset (Eye Gestures) 1 Sample Count	5	5	250	250 x 5
2nd EEG Dataset (Eye Gestures) 2 Sample Count	10	5	250	250 x 10

3.4 The EEG Datasets

The EEG datasets are divided into 2 categories; 2 classification data, Neutral and Smile; and 4 classification data, all face gestures altogether. One additional parameter present for the ANFIS algorithm was the number of epochs, which was set to 5 on the first run, and 10 on the second run. The EEG datasets are obtained with 1, 2, and 4

sample counts per block. Sample counts refer to the number of samples a particular sensor obtains in an epoch. For 1 sample count, the 5 sensors obtained 1 sample each, consequently producing a dataset with 5 features. The 2 sample counts per block obtained 2 samples for each sensor, consequently producing a dataset with 10 features. Finally, the 4 sample counts per block obtained 4 samples for every sensor, which produces a dataset of 20 features. Overall, there were a total of 6 EEG datasets in this set, that are to be fed to the algorithm. The conditions implemented to the 1st EEG dataset was also implemented to the 2nd EEG dataset. Table 4 below shows the size of the datasets used in the experiments.

4 Results and Discussion

In this section, both the ANFIS and SVM algorithms were fed with the obtained datasets and evaluated with the performance metrics of accuracy, prediction time, and training time. The tests are conducted 5 times and an average was obtained from all the runs. This should provide sufficient information in determining the performance of ANFIS together with a side-by-side comparison with the SVM.

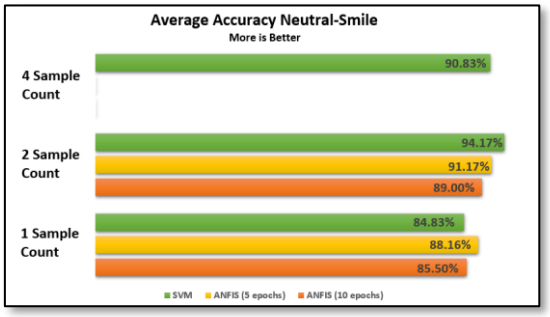

Fig. 12. 1st EEG Dataset - Average Accuracy of the 3 Algorithms (Neutral - Smile Classification)

4.1 Simulation Results for 1st EEG Dataset

Figure 12 shows the results of the algorithms when fed with the 1st EEG dataset, Face Gestures, with 2 classifications, namely Neutral and Smile. Additionally, the dataset had variations on sample counts, 1, 2, and 4, which consequently added more features into the dataset, shown in Table 4. Each dataset variation consisted of 200 rows of data, around 100 samples for each classification. In the 1 sample count, all the algorithms were able to get an accuracy rating above 80%, with ANFIS (5 Epochs) as the highest with 88.16%. This was followed by ANFIS (10 Epochs) with an accuracy rating of 85.50%, and finally the SVM with 84.83%. On 2 sample count, the algorithms showed better performance compared to the performance on the dataset with 1 sample count. The SVM showed the most significant improvement with an accuracy rating of 94.17%. ANFIS on the other hand showed at least a 3% improvement by obtaining a

performance rating of 91.17% and 89.00% for 5 and 10 epochs respectively. Finally, on the dataset with 4 sample counts, only the SVM was able to produce an output with a reading of 90.83% for its accuracy, while the ANFIS algorithm crashed due to memory error.

This experiment validated and consolidated 2 ideas, namely that

(i) the performance of the ANFIS algorithm is indeed comparable to the performance of the SVM

(ii) increasing the sample count in the dataset offers a degree of improvement in the performance of the algorithms

However, on the dataset with 4 sample counts, the ANFIS algorithm failed to generate predictions as it has reached a 'memory error.' This implied that the ANFIS algorithm ran out of RAM to be able to train itself and create predictions, given this dataset. Focusing on the results for the 1 and 2 sample counts, it is observed that ANFIS marginally outperforms the SVM by at least 1%; but, in the dataset of 2 sample counts, the SVM greatly benefitted from the increase in the number of features and outperformed the ANFIS by at least 3%. Though generally, the difference in performance between the 2 algorithms is not significant, it did present a validation on the ideas presented at the start of this paragraph.

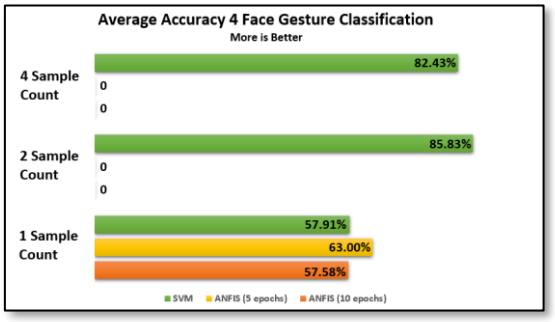

Fig. 13. 1st EEG Dataset - Average Accuracy of the 3 Algorithms (4 Face Gesture Classification)

Immediately shown in Figure 13 is that the ANFIS was not able to produce any predictions on the dataset with 2 and 4 sample counts. The dataset fed on this round of experimentation possessed 800 rows of input, 200 for each classification. The SVM was able to produce predictions with 85.83% and 82.43% accuracy with the dataset consisting of 2 sample counts and 4 sample counts respectively. On the dataset with 1 sample count, the algorithms performed significantly poorer as compared to the previous test. The SVM generated predictions with a 57.91% accuracy rating, while the ANFIS predicted with accuracies of 57.58% and 63.00% for 5 and 10 epochs respectively.

This test presented another set of observations. It would seem that the ANFIS algorithm was incapable of managing large datasets. By observing the algorithm at work,

the researcher observed that on the forward-pass, the ANFIS runs out of memory and crashes, considering that the computing hardware possesses 16 Gb of memory.

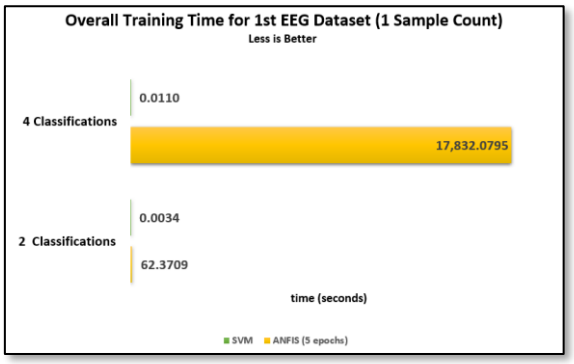

Fig. 14. 1st EEG Dataset – Overall Training Time for SVM and ANFIS

Another observation is that 2 sample counts in the dataset was sufficient to produce significantly more accurate results as observed in the results of SVM's performance on this test and from the previous test. Focusing on the results on the dataset with 1 sample count, the ANFIS (10 Epochs) performed relatively on par with the SVM while ANFIS with 5 epochs performed marginally better on both tests. This infers that on this type of dataset, the ANFIS performs better with 5 epochs.

Table 5. 1st EEG Dataset Overall Algorithm Training Time

No. of Sample Counts	ANFIS (5 Epoch)		ANFIS (10 Epoch)		SVM	
	2 Classes	4 Classes	2 Classes	4 Classes	2 Classes	4 Classes
1 Sample Count	62.37 s	17,832 s	144.08 s	37,025 s	0.0034 s	0.0110 s
2 Sample Count	36,954 s	-	77,295 s	-	0.0032 s	0.0088 s
4 Sample Count	-	-	-	-	0.0120 s	0.0653 s

Considering the phenomenon observed with the ANFIS algorithm exhibited when managing relatively large datasets, Figure 14 presents the time it took for both ANFIS and SVM to train. Note that only ANFIS with 5 epochs is considered in this comparison as results from 10 epochs would roughly be double the value of the results from the 5 epochs. Another note was that, for this comparison, only the results from the dataset with 1 sample count were used as it offers a fair comparison between the 2 algorithms. It was observed that the duration SVM takes for training was only a small fraction of a second, 0.0034s, and 0.0110s specifically for 2 classifications and 4 classifications respectively. ANFIS, on the other hand, took significantly longer for training with 62.3709s for 2 classifications, and 17,832.0795s for 4 classifications. The latter is equivalent to 4.95 hours. This inferred that the ANFIS took up significantly more

computational resources than the SVM for this use-case. Table 5 shows the complete results of the training duration in seconds for all algorithms and datasets.

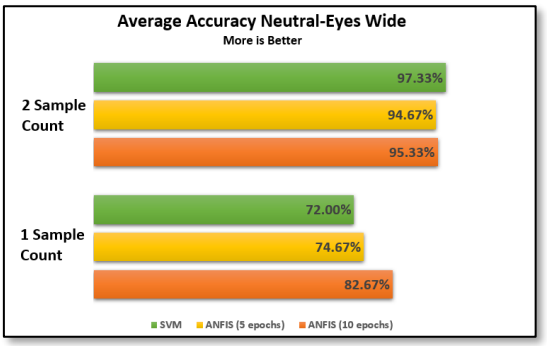

Fig. 15. 2nd EEG Dataset – Average Accuracy of the 3 Algorithms (Neutral – Eyes Wide Classification)

4.2 Simulation Results for 2nd EEG Dataset

Figure 15 showed the results for the 2nd EEG Dataset with 2 classifications. The dataset was composed of 5 features and 10 features for 1 and 2 sample counts respectively. There were 50 rows of input information for each classification, summing up to a total of 100 rows for this dataset. The results obtained from the dataset with 1 sample count showed satisfactory results. The SVM was able to obtain a 72.00% accuracy rating while the ANFIS obtained 74.67% and 82.67% for 5 and 10 epochs respectively. The same phenomenon from the 1st EEG Dataset was observed here on the results from the dataset with 2 sample counts. The SVM experienced the most benefits with the added features with an accuracy rating of 97.33%. In addition, the ANFIS algorithm also experienced significant improvements with the added features, gaining average accuracy ratings of 94.67% and 95.33% for 5 and 10 epochs respectively.

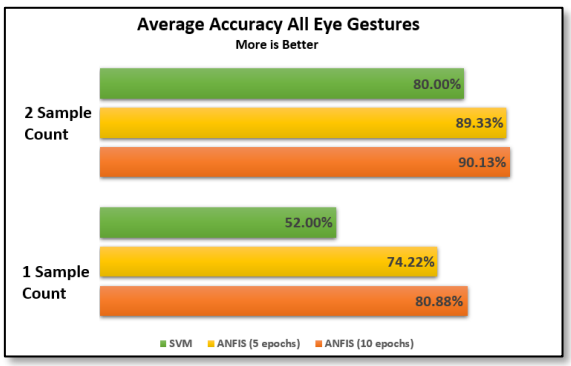

Fig. 16. 2nd EEG Dataset – Average Accuracy of the 3 Algorithms (All Eye Gesture Classification)

In this set of EEG data, it can be observed that the ANFIS performed significantly better as compared to the 1st set; this may be due to the smaller sized dataset fed to the algorithm. Though the SVM outperforms the ANFIS on the dataset with 2 sample counts, the ANFIS only lags by 2.66%, which was still very satisfactory. The same phenomenon from the 1st EEG dataset was observed on the dataset with 1 sample count, where ANFIS still outperformed the SVM. One thing to note was that for both sample count categories, the ANFIS 10 epoch performed better than the ANFIS running on 5 epochs. This inferred that, on this dataset, 10 epochs were better than 5 epochs.

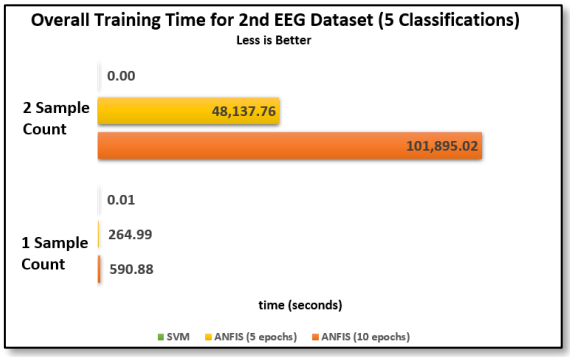

Fig. 17. 2nd EEG Dataset – Overall Training Time of the 3 Algorithms (5 Classifications)

Figure 16 represented the results when the dataset all 5 classifications were fed to the algorithms. For consistency, each classification held 50 rows of information, totaling up to 250 rows in the dataset. The SVM had an average performance while generating predictions with the 1 sample count dataset, obtaining an accuracy rating of 52.00%. ANFIS on the other hand performed satisfactorily with accuracy ratings of 74.22% and 80.88% for 5 and 10 epochs respectively. For the dataset with 2 sample counts, all the algorithms performed satisfactorily with at least an 80.00% accuracy rating for the SVM. The ANFIS on the other hand was able to obtain 89.33% running for 5 epochs 90.13% running for 10 epochs.

Every test so far had consistent results, such as SVM experiencing a significant improvement in accuracy in the presence of more features and ANFIS performing better with the 1 sample count dataset. However, it could be observed in this test that the ANFIS performed better on the dataset with 2 sample counts. As observed in the 1st EEG dataset, a degree of accuracy drop was expected in creating predictions with datasets having 2 classifications to 5 classifications. Compared to the SVM, the ANFIS did not suffer a much accuracy drop with the highest observable drop to be 5.34%, while the highest observable accuracy drops the SVM experienced is 20%. This showed that the ANFIS tends to be more consistent in managing this dataset when introducing more classifications.

According to the long training time which is required with the 2nd EEG dataset, we reasonably decided to focus on the dataset with 5 classifications. Figure 17 shows the

time in seconds it took to train the SVM, ANFIS (5 Epochs), and ANFIS (10 Epochs). Consistently, the SVM only took a fraction of a second for training and providing satisfactory results for both 1 and 2 sample counts. ANFIS on the other hand took significantly longer to train in the dataset with 1 sample count, with 5 epochs taking 264.99 seconds or 4.42 minutes, and the 10 epochs taking roughly double the time with that of 5 epochs. Interestingly, there was an exponential growth in the duration of training for ANFIS with the dataset with 2 sample count, introducing double the number of features. For 5 epochs, the ANFIS took 48,137.76 seconds to train, roughly 13.37 hours, and 10 epochs roughly double the amount, as expected. This phenomenon was also observed in the training with the 1st EEG dataset. Considering that the 2nd EEG dataset is significantly smaller than the previous dataset, the training time required for the ANFIS algorithm still experienced an exponential increase when more features are introduced in the dataset. Table 9 shows the results for the time duration it took to train the algorithms with the 2nd EEG dataset.

Table 6. 2nd EEG Dataset - Overall Algorithm Training Time

No. of Samples	ANFIS (5 Epoch)		ANFIS (10 Epoch)		SVM	
	2 Class	5 Class	2 Class	5 Class	2 Class	5 Class
1 Sample Count	15.11 s	264.99 s	33.88 s	590.88 s	0.0020 s	0.0054 s
2 Sample Counts	13,540 s	48,137 s	28,898 s	101,895 s	0.0020 s	0.0037 s

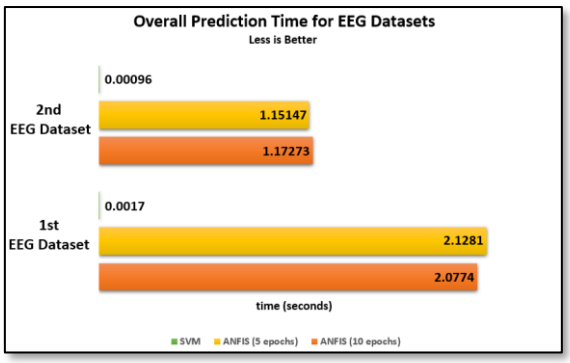

Fig. 18. Overall Algorithm Prediction Time

4.3 Prediction Time for Both Algorithms

Figure 18 showed the prediction time for the algorithms. It can be seen that the SVM can generate predictions almost instantaneously, while the ANFIS took around 2 seconds to generate predictions when working with the 1st EEG Dataset, and around 1 second when working with the 2nd EEG Dataset. This result suggests that the ANFIS was slower than the SVM, but during the test flight experiment, there were no observable delays in maneuvering the quadcopter.

4.4 Summary of Research Findings

During the tests, both the SVM and ANFIS exhibited similar performance in terms of accuracy. However, it was observed with the 1st EEG Dataset, the ANFIS seemed to require a lot of computational resources as it failed to generate predictions with the datasets having 2 and 4 sample counts. Suspicious with the size of the dataset, the researcher gathered the 2nd set of EEG data which was significantly smaller than the first dataset. When this dataset was fed to the ANFIS algorithm, the algorithm was able to generate a generalization and was able to accurately predict, exhibiting a satisfactory performance; but, the ANFIS algorithm took a lot of computational time to train when more features were introduced. This confirmed 2 ideas, namely that (i) the ANFIS algorithm takes up a lot of computational resources with respect to the size of the data being fed, and that (ii) the ANFIS is suitable for a smaller dataset. A study conducted in [15], the researchers mentioned that the ANFIS was effective when dealing with around 6 features in the dataset. The findings in this research have supported that claim as observed in the simulation tests for both datasets.

Analyzing the architecture of the ANFIS algorithm, it was observed that with a dataset of n features and m classifications, the algorithm will not only generate $n \times m$ number of membership functions but will calibrate the weights of each membership function by the forward and backward pass sequence of the algorithm; and results from this study showed that the algorithm can take as much as 28 hours for training alone.

The benefit of this process is that the ANFIS was able to generate predictions that are relatively more precise compared to the SVM. Since the algorithm took the mean and standard deviation of every feature of every classification it was able to generate membership functions that for a particular classification that was unlikely to be affected when including the data for the additional classification. Users can extend the dataset by adding more classifications and the ANFIS may experience an acceptable accuracy drop as observed in the simulation tests with the 2nd EEG Dataset. In terms of accuracy and reliability, the ANFIS was able to perform as well as the SVM, or even marginally better, however, the SVM was more efficient than the ANFIS algorithm as the latter takes up significant amounts of computational resources.

5 Conclusion and Recommendation

This research explored the applicability of a Neuro-Fuzzy Algorithm, specifically the Adaptive Neuro-Fuzzy Inference System (ANFIS), in a BCI system implementations. The main goal of this research was to analyze the performance of the ANFIS algorithm, together with the SVM, to manage EEG datasets.

The developed ANFIS algorithm was compared against the SVM algorithm, a known and popular algorithm serving as a baseline comparison, in terms of accuracy, training, and prediction time. The research employed the use of facial gestures to generate EEG signals which are captured by 2 main sensors *AF3* and *AF4* of the Emotiv INSIGHT. The face gestures Neutral, Smile, Shocked, and Clench were all used as the gestures of the 1st EEG dataset. The captured EEG signals were sent to the computing

hardware where the dataset was recorded and fed to the developed algorithms. Initial simulation tests showed that the ANFIS was capable of performing nearly as well as the SVM; however, the prior was not able to process large datasets, returning a memory error. This error was determined to be a limitation in the computing hardware. After analyzing the results of the simulation tests with the 1st EEG dataset, it was observed that the ANFIS took up a lot of computational resources, specifically memory; and, the ANFIS significantly took more training time compared to the SVM; additionally, the dataset was found to be too noisy, which was one of the causes of poor performance.

The researcher then proceeded to collect another set of EEG data with the use of eye gestures instead of facial gestures. The eye gestures used are neutral, eyes opened-wide, left wink, right wink, and eyes shut, this composed the 2nd EEG dataset. The obtained EEG data was then filtered with a 5th order Butterworth band-pass filter with a lower cut-off frequency of 13 Hz and a higher cut-off frequency of 43 Hz. This removes the brainwaves which are sensitive to external interference. The filtered data was then recorded and further processed heuristically by determining the minimum and maximum points of the instance of the eye gesture, selecting points of interest, and recording accordingly. This has left the researcher with a significantly smaller dataset which was subject to the same sequence of experiments.

Results from the simulation tests of the 2nd EEG dataset showed significant improvements from both algorithms by obtaining accuracy ratings of 90.13% for the ANFIS algorithm and 80.00% for the SVM algorithm while working with the dataset with all eye gesture classifications and 2 Sample Counts. ANFIS still suffered from long training duration as high as 28.3 hours. It was later observed that this high computational resource requirement was due to the architecture of ANFIS, wherein each feature possessed 5 membership functions, totaling up to 50 membership functions. It took the algorithm a substantial amount of time to train these membership functions on 10 epochs. It was also observed that the ANFIS performed more precisely than of the SVM, which again was discovered to be due to its architecture. The ANFIS algorithm in this research employed a Gaussian membership function which obtained the mean and standard deviation from each feature from each classification, generating a set of membership functions, tailored fit to a particular classification.

Based on the results, it can be concluded that the ANFIS is a viable algorithm for a BCI system implementation as its accuracy ratings are comparable to the SVM. However, users must be cautious of the data to be fed to the ANFIS algorithm as the algorithm is more suited for a smaller dataset. Feeding the ANFIS with large datasets, especially datasets with a lot of features will require large amounts of computational resources. Researchers can utilize the methodology presented in this article and utilize the Python Scripting Box from OpenViBE to interface with robots while using the Python programming language. However, the researcher has noticed that the Python Box function of the software only supports Python 2. This presented the researcher with various limitations, particularly the incompatibility of known Machine Learning libraries such as TensorFlow, Keras, and PyTorch. A direct interface between the neuroheadset and pre-processing algorithms coded in Python 3 may yield better results for future researchers on this topic.

Research findings are open to other methodologies were intuitiveness of the interfaces can support the end-user while interacting with Machine Interface [16-19].

Acknowledgments

This work was presented in dissertation form in fulfilment of the requirements for the MSc in Robotics Engineering for the student Timothy Chu under the supervision of E.L. Secco from the Robotics Laboratory, School of Mathematics, Computer Science and Engineering, Liverpool Hope University, UK and Dr Alvin Chua from the Mechanical Engineering Department, De La Salle University, PH.

References

1. Prince, D., Edmonds, M., Sutter, A., Cusumano, M., Lu, W. and Asari, V.: Brain machine interface using Emotiv EPOC to control robai cyton robotic arm. In 2015 National Aerospace and Electronics Conference (NAECON) (pp. 263-266). IEEE (2015).
2. Holewa, K. and Nawrocka, A.: Emotiv EPOC neuroheadset in brain-computer interface. In Proceedings of the 2014 15th International Carpathian Control Conference (ICCC) (pp. 149-152). IEEE (2014).
3. Li, S. and Feng, H.: EEG Signal Classification Method Based on Feature Priority Analysis and CNN. In 2019 International Conference on Communications, Information System and Computer Engineering (CISCE) (pp. 403-406). IEEE (2019).
4. Zeng, R., Bandi, A. and Fellah, A.: Designing a Brain Computer Interface Using EMOTIV Headset and Programming Languages. In 2018 Second International Conference on Computing Methodologies and Communication (ICCMC) (pp. 908-913). IEEE (2018).
5. Aguiar, S., Yanez, W. and Benítez, D.: Low complexity approach for controlling a robotic arm using the Emotiv EPOC Headset. In 2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) (pp. 1-6). IEEE (2016).
6. Mamani, M.A. and Yanyachi, P.R.: Design of computer brain interface for flight control of unmanned air vehicle using cerebral signals through headset electroencephalograph. In 2017 IEEE International Conference on Aerospace and Signals (INCAS) (pp. 1-4). IEEE (2017).
7. Chiuzbaian, A., Jakobsen, J. and Puthusserypady, S.: Mind Controlled Drone: An Innovative Multiclass SSVEP based Brain Computer Interface. In 2019 7th International Winter Conference on Brain-Computer Interface (BCI) (pp. 1-5). IEEE (2019).
8. Jang, J.S.: ANFIS: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics, 23(3), pp.665-685 (1993).
9. Tavakoli, M., Benussi, C., Lopes, P.A., Osorio, L.B. and de Almeida, A.T.: Robust hand gesture recognition with a double channel surface EMG wearable armband and SVM classifier. Biomedical Signal Processing and Control, 46, pp.121-130 (2018).
10. Spüler, M.: A Brain-Computer Interface (BCI) system to use arbitrary Windows applications by directly controlling mouse and keyboard. In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 1087-1090). IEEE (2015).
11. How Your Brain Works, https://science.howstuffworks.com/life/inside-the-mind/human-brain/brain8.htm, last accessed 2020/09/11.

12. Lobes Of The Brain, https://qbi.uq.edu.au/brain/brain-anatomy/lobes-brain, last accessed 2020/09/29.
13. The Introductory Guide To EEG (Electroencephalography) – EMOTIV, https://www.emotiv.com/eeg-guide/, last accessed 2020/09/20.
14. Filtering – The Basics, https://blricrex.hypotheses.org/filtering-introduction#Filtering%20in%20the%20time%20domaine, last accessed 2020/09/25
15. Jahankhani, P., Revett, K. and Kodogiannis, V.: A rule based approach to classification of EEG datasets: a comparison between ANFIS and rough sets. In 2008 9th Symposium on Neural Network Applications in Electrical Engineering (pp. 157-160). IEEE (2008).
16. D. Elstob, E.L. Secco, A low cost EEG based BCI Prosthetic using motor imagery, (2016), International Journal of Information Technology Convergence and Services, vol. 6, n. 1, pp. 23-36, http://arxiv.org/abs/1603.02869
17. E.L. Secco, C. Moutschen, A. Tadesse, M. Barrett-Baxendale, D. Reid, A. Nagar, Development of a sustainable and ergonomic interface for the EMG control of prosthetic hands, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 192, 321-327, 2017, Springer, ISBN 978-3-319-58877-3
18. E.L. Secco, P. Caddet, A.K. Nagar, Development of an Algorithm for the EMG Control of Prosthetic Hand, Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing, 1139, chapter 15, Springer, DOI: 10.1007/978-981-15-3287-0_15
19. A.T. Maereg, Y. Lou, E.L. Secco, R. King, Hand Gesture Recognition Based on Near-Infrared Sensing Wristband, Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020), 110-117, 2020 - ISBN: 978-989-758-402-2 – DOI: 10.5220/0008909401100117